UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/813,615	03/31/2004	Simon Knowles	321547.US	3818

27964 7590 07/07/2010

HITT GAINES P.C.
P.O. BOX 832570
RICHARDSON, TX 75083

EXAMINER	
FENNEMA, ROBERT E	

ART UNIT	PAPER NUMBER
2183	

NOTIFICATION DATE	DELIVERY MODE
07/07/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket@hittgaines.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/813,615
Filing Date: March 31, 2004
Appellant(s): KNOWLES, SIMON

J. Joel Justiss (Reg. No. 48,981)
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 4/23/2010 appealing from the Office action mailed 8/4/2009.

(1) Real Party in Interest

The examiner has no comment on the statement, or lack of statement, identifying by name the real party in interest in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The following is a list of claims that are rejected and pending in the application:

1-6, 8-18, and 21

(4) Status of Amendments After Final

The examiner has no comment on the appellant's statement of the status of amendments after final rejection contained in the brief.

(5) Summary of Claimed Subject Matter

The examiner has no comment on the summary of claimed subject matter contained in the brief.

(6) Grounds of Rejection to be Reviewed on Appeal

The examiner has no comment on the appellant's statement of the grounds of rejection to be reviewed on appeal. Every ground of rejection set forth in the Office action from which the appeal is taken (as modified by any advisory actions) is being maintained by the examiner except for the grounds of rejection (if any) listed under the subheading "WITHDRAWN REJECTIONS." New grounds of rejection (if any) are provided under the subheading "NEW GROUNDS OF REJECTION."

(7) Claims Appendix

The examiner has no comment on the copy of the appealed claims contained in the Appendix to the appellant's brief.

(8) Evidence Relied Upon

5423051	Fuller et al.	6-1995
20050044434	Kahle et al.	2-2005
20040215593	Sharangpant et al.	10-2004
5922065	Hull et al.	7-1999

- Bolotski, Michael et al. "Unifying FPGAs and SIMD Arrays". MIT: 1994.

- Stokes, Jon. "A Brief Look at the PowerPC 970". Ars technica: 2002. 11 May, 2006.

- Hennessy, John. Patterson, David. "Computer Architecture: A Quantitative Approach". Unknown Edition, May 2003. Pages 127-130 and D-1 through D-14.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-7, 11, 14-18, and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hull, in view of “Unifying FPGAs and SIMD Arrays” by Bolotski et al. (herein Bolotski), further in view of Hennessy et al. ("Computer Architecture: A Quantitative Approach", herein Hennessy).

3. As per Claim 1, Hull teaches: A computer processor for processing (i) instruction packets comprising a plurality of only control instructions (Figure 4, Template “B”), the control instructions having a control bit width (inherent), and (ii) instruction packets comprising a plurality of instructions comprising at least one data processing instruction (Figure 4, for example, Template “E, the processor comprising:

a decode unit for decoding sequentially the instruction packets fetched from a memory holding a sequence of instruction packets (Column 5, Lines 16-18);

a control processing channel capable of performing control operations, the control processing channel comprising a plurality of functional units (Column 3, Line 51,

the branch execution unit) including a control register file having a first bit width (Column 3, Lines 18-21, 64 bits); and

a data processing channel capable of performing data processing operation, the data processing channel comprising a plurality of functional units (Column 3, Lines 48-50, the integer, memory, and floating point execution units) including a data register file having a second bit width wider than the first bit width (Column 3, Lines 9-11, the floating point registers have a bit width of 82 bits);

wherein the decode unit comprises decode circuitry configured to decode identification bits of each instruction packet to determine which type (i), (ii) of instruction packet is being decoded, and control circuitry configured to pass the plurality of only control instructions from an instruction packet of type (i) to the control processing channel when the decode circuitry indicates so and to pass the plurality of instructions comprising at least one data processing instruction from an instruction packet of type (ii) to the data processing channel when the decode circuitry indicates so (Column 3, Lines 63-66, the template field in each packet/bundle tells the machine what is in each packet, and directs the packet to the appropriate functional units. Template “B” for example contains 3 branch (control) instructions, and Template “0” contains only data-processing instructions);

wherein, in use the decode unit causes instructions of (i) instruction packets comprising a plurality of only control instructions to be executed sequentially on the control processing channel (Column 4, Lines 61-62); and

wherein in use the decode unit causes instructions of (ii) instruction packets comprising a plurality of instructions comprising at least one data processing instruction to be executed simultaneously on the data processing channel (Column 2, Lines 5-9), but fails to teach:

at least one input of which is a vector, and

the data processing instructions having a data processing bit width wider than the control bit width.

Hull is silent towards vectors (using the Applicant's definition of a vector in Paragraph 23 of the specification, wherein a vector is an assembly of scalar operands). Hull does however teach that there is a high demand for highly efficient, parallel processing machines, and Bolotski teaches that a massively parallel system of computing is a SIMD processor, which is a vector, or array, of scalar values. Given the disclosed need in Hull of increased parallelism, the teachings of both Bolostki of the parallelism of SIMD (along with SIMD being well known in the art), one of ordinary skill in the art would have been motivated to integrate SIMD processing into the system of Hull, to further exploit parallelism in lines with the goals of Hull.

However, Hull and Bolostki are silent towards the data processing instructions being wider (longer) than a control instruction, because Hull teaches a fixed-length architecture. However, Hull does note in Column 1, Lines 15-23 that previous instruction sets used variable-length architecture, but Hull happens to be designed for a RISC machine. Hennessy (a textbook) teaches the differences between a variable and fixed

architecture, and shows that variable-length instructions create smaller programs (Page 128), and that on average, the instructions are significantly smaller than RISC instructions (Page 128 and Page 129, Figure 2.23) because they do not require wasted and unneeded fields. Given that Hull's invention is designed to reduce waste and inefficiency in code size (Column 2, Lines 3-5), and to help avoid the expanded code caused by fixed-length instructions (Column 1, Lines 52-60), one of ordinary skill in the art would have been motivated to implement Hull's invention in a variable-length instruction set, to further increase the flexibility of the code, and reduce its size, while still taking advantage of the alignment and efficiencies provided by Hull's templates.

Given that combination, Examiner notes Page D-13 of Hennessy, which shows that the control instruction (a, JE), has a smaller length than at least two data processing instructions (c, MOV and e, ADD) in these examples.

4. As per Claim 2, Hull teaches: A computer processor according to claim 1, wherein the control processing channel further comprises a branch unit and a control execution unit (Colum

5. As per Claim 3, Hull teaches The computer processor according to claim 1, wherein the data processing channel further comprises a fixed data execution unit (Column 3, Lines 48-51), but fails to teach:

the second processing channel containing a configurable data execution unit.

While Hull teaches a fixed data execution unit, Hull is silent towards a configurable data execution unit. However, Bolotski teaches a system that can simulate SIMD and configurable operations on the same unit (Section 4), which can be subdivided until SIMD and configurable units. Boloski further teaches that the advantages of combining a SIMD and configurable unit includes reducing cost by not duplicating logic (Section 4.1), additionally, one of ordinary skill in the art would recognize the additional benefits of configurable hardware, such as being able to configure exactly the units you need, allowing for flexibility and increased performance. Therefore, one of ordinary skill in the art would have been motivated to combine the teachings of Hull and Bolotski to implement configurable data execution units.

6. As per Claim 4, Bolotski teaches: A computer processor according to claim 3, wherein the fixed data execution unit and the configurable data execution unit both operate according to a single instruction multiple data format (Section 4).

7. As per Claim 5, Hull teaches: The computer processor according to claim 1, wherein the control and data processing channels share a load store unit (Column 3, Line 50, there is only one memory execution unit, and since the branch unit needs to be able to have access to memory (to fetch the new target instructions), it clearly has to be shared between the two).

8. As per Claim 6, Hull teaches: A computer processor according to claim 5, wherein the load store unit uses control information supplied by the control processing channel and data supplied by the data processing channel (Column 3, Line 50, there is only one memory execution unit, the data processing channel needs to use it to fetch data from memory, and the control channel needs to use it to fetch branch target addresses).

9. As per Claim 7, Hull teaches: A computer processor according to claim 1, wherein the instruction packets are all of equal bit length (Column 2, Lines 22-24, 128 bit bundles).

10. As per Claim 11, Hull teaches: A computer processor according to claim 7, wherein the nature of each instruction in an instruction packet is selected at least from a control instruction, a data instruction, and a memory access instruction (Column 3, Lines 48-51, also see Figure 4).

11. As per Claim 14, Hull teaches: A computer processor according to claim 1, wherein when the decode unit detects that the instruction packet defines three control instructions, the decode unit is operable to supply the control processing channel with the three control instructions (Column 3, Lines 65-66) whereby the three control instructions are executed sequentially (Column 4, Lines 61-62).

12. As per Claim 15, Hull teaches: A computer processor according to claim 1, wherein when the decode unit detects that the instruction packet defines two instructions comprising at least one data instruction, the decode unit is operable to supply the second processing channel with at least the data instruction (Column 3, Lines 65-66) whereby the two instructions are executed simultaneously (Column 2, Lines 5-9).

13. As per Claim 16, Hull teaches: A computer processor according to claim 1, wherein the decode unit is operable to read the values of a set of designated bits at predetermined bit locations in each instruction packet of the sequence (Column 3, Lines 63-66, the template bits), to determine:

a) whether the instruction packet defines a plurality of only control instructions (Figure 4, Template “B”) or a plurality of instructions of which at least one is a data instruction (Figure 4, any non-“B” template); and

b) where the instruction packet defines a plurality of instructions of which at least one is a data instruction, the nature of each of the two instructions selected from: a control instruction; a data instruction; and a memory access instruction (Figure 4, see Template “0” or “8” for example).

14. As per Claim 17, Hull teaches: A computer processor according to claim 3, wherein the configurable data execution unit is capable of executing more than two

consecutive operations on the data provided by a single issued instruction before returning a result to a destination register file.

One of ordinary skill in the pertinent art would have recognized that this is a simple accumulate function that would be easily programmable in configurable logic.

15. As per Claim 18, Hull teaches: A method of operating a computer processor for processing (i) instruction packets comprising a plurality of only control instructions (Figure 4, Template "B"), the control instructions having a control bit width (inherent); and (ii) instruction packets comprising a plurality of instructions comprising at least one data processing instruction (Figure 4, Template "0"),

the processor comprising a decode unit for decoding sequentially the instruction packets fetched from a memory holding the instruction packets (Column 5, Lines 16-18):

a control processing channels comprising a plurality of functional units (Column 3, Line 51, the branch execution unit) including a control register file having a first bit width (Column 3, Lines 18-21, 64 bits) and a data processing channel comprising a plurality of functional units including a data register file having a second bit width, wider than the first bit width (Column 3, Lines 9-11, the floating point register file has a bit width of 82 bits), the method comprising:

decoding identification bits of each instruction packet to determine which type (i), (ii), of instruction packet is being decoded, and passing the plurality of only control instructions from an instruction packet of type (i) to the control processing channel when

the decode circuitry indicates so and passing the plurality of instructions comprising at least one data processing instruction from an instruction packet of type (ii) to the data processing channel when the decode circuitry indicates so (Column 3, Lines 63-66, the template field in each packet/bundle tells the machine what is in each packet, and directs the packet to the appropriate functional units. Template “B” for example contains 3 branch (control) instructions, and Template “0” contains only data-processing instructions);

when the instruction packet defines (i) a plurality of only control instructions, supplying the control instructions to the control processing channel wherein the control instructions are executed sequentially (Column 4, Lines 61-62); and

when the instruction packet defines (ii) a plurality of instructions comprising at least one data processing instruction, supplying at least the data instruction to the data processing channel wherein the plurality of instructions are executed simultaneously (Column 2, Lines 5-9), but fails to teach:

the data processing instructions having a data processing bit width wider than the control bit width; and

a data processing channel capable of performing data processing operations at least one input of which is a vector.

Hull is silent towards vectors (using the Applicant's definition of a vector in Paragraph 23 of the specification, wherein a vector is an assembly of scalar operands). Hull does however teach that there is a high demand for highly efficient, parallel processing machines, and Bolotski teaches that a massively parallel system of

computing is a SIMD processor, which is a vector, or array, of scalar values. Given the disclosed need in Hull of increased parallelism, the teachings of both Bolostki of the parallelism of SIMD (along with SIMD being well known in the art), one of ordinary skill in the art would have been motivated to integrate SIMD processing into the system of Hull, to further exploit parallelism in lines with the goals of Hull.

However, Hull and Bolostki are silent towards the data processing instructions being wider (longer) than a control instruction, because Hull teaches a fixed-length architecture. However, Hull does note in Column 1, Lines 15-23 that previous instruction sets used variable-length architecture, but Hull happens to be designed for a RISC machine. Hennessy (a textbook) teaches the differences between a variable and fixed architecture, and shows that variable-length instructions create smaller programs (Page 128), and that on average, the instructions are significantly smaller than RISC instructions (Page 128 and Page 129, Figure 2.23) because they do not require wasted and unneeded fields. Given that Hull's invention is designed to reduce waste and inefficiency in code size (Column 2, Lines 3-5), and to help avoid the expanded code caused by fixed-length instructions (Column 1, Lines 52-60), one of ordinary skill in the art would have been motivated to implement Hull's invention in a variable-length instruction set, to further increase the flexibility of the code, and reduce its size, while still taking advantage of the alignment and efficiencies provided by Hull's templates.

Given that combination, Examiner notes Page D-13 of Hennessy, which shows that the control instruction (a, JE), has a smaller length than at least two data processing instructions (c, MOV and e, ADD) in these examples.

16. As per Claim 21, Hull teaches: A computer-readable medium comprising a sequence of instruction packets (Column 2, Lines 22-24),

said instruction packets including a first type of instruction packet comprising a plurality of only control instructions of equal width (Figure 4, Template “B”), the control instructions having a control bit width (inherent), and a second type of instruction packet comprising a plurality of instructions including at least one data processing instruction (Figure 4, Template “0”), said instruction packets including at least one indicator bit at a designated bit location within the instruction packet (Column 3, Lines 63-66, the template fields), wherein the computer-readable medium is adapted to run on a computer such that said indication bit is adapted to cooperate with a decode unit of the computer to designate whether:

a) the instruction packet defines a plurality of only control instructions or a plurality of instructions comprising at least one is a data instruction (Column 3, Lines 63-66); and

b) in the case when there is a plurality of instructions comprising at least one data instruction, the nature of each of the first and second instructions selected from: a control instruction; a data instruction; and a memory access instruction (Figure 4, see Templates "0" or "8" for example), but fails to teach:

the at least one data processing instructions having a data processing bit width wider than the control bit width, and wherein at least one data processing instruction is a vector.

Hull is silent towards vectors (using the Applicant's definition of a vector in Paragraph 23 of the specification, wherein a vector is an assembly of scalar operands). Hull does however teach that there is a high demand for highly efficient, parallel processing machines, and Bolotski teaches that a massively parallel system of computing is a SIMD processor, which is a vector, or array, of scalar values. Given the disclosed need in Hull of increased parallelism, the teachings of both Bolostki of the parallelism of SIMD (along with SIMD being well known in the art), one of ordinary skill in the art would have been motivated to integrate SIMD processing into the system of Hull, to further exploit parallelism in lines with the goals of Hull.

However, Hull and Bolostki are silent towards the data processing instructions being wider (longer) than a control instruction, because Hull teaches a fixed-length architecture. However, Hull does note in Column 1, Lines 15-23 that previous instruction sets used variable-length architecture, but Hull happens to be designed for a RISC machine. Hennessy (a textbook) teaches the differences between a variable and fixed architecture, and shows that variable-length instructions create smaller programs (Page 128), and that on average, the instructions are significantly smaller than RISC instructions (Page 128 and Page 129, Figure 2.23) because they do not require wasted and unneeded fields. Given that Hull's invention is designed to reduce waste and inefficiency in code size (Column 2, Lines 3-5), and to help avoid the expanded code caused by fixed-length instructions (Column 1, Lines 52-60), one of ordinary skill in the art would have been motivated to implement Hull's invention in a variable-length

instruction set, to further increase the flexibility of the code, and reduce its size, while still taking advantage of the alignment and efficiencies provided by Hull's templates.

Given that combination, Examiner notes Page D-13 of Hennessy, which shows that the control instruction (a, JE), has a smaller length than at least two data processing instructions (c, MOV and e, ADD) in these examples.

17. Claims 8-10, 12 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hull, Hennessy, and Bolotski, in view of In re Rose.

18. These claims recite limitations of the bit lengths of various instructions and packets.

While Hull may not teach the recited lengths, if it was advantageous to lengthen or shorten the bit lengths in Hull for various reasons, one of ordinary skill in the pertinent art would have recognized that it would have been simple to do so.

Further, it has been found that a change in size does not produce a patentable distinction. In re Rose , 220 F.2d 459, 105 USPQ 237 (CCPA 1955) (Claims directed to a lumber package “of appreciable size and weight requiring handling by a lift truck” where held unpatentable over prior art lumber packages which could be lifted by hand because limitations relating to the size of the package were not sufficient to patentably distinguish over the prior art.); In re Rinehart, 531 F.2d 1048, 189 USPQ 143 (CCPA 1976) (“mere scaling up of a prior art process capable of being scaled up, if such were

the case, would not establish patentability in a claim to an old process so scaled." 531 F.2d at 1053, 189 USPQ at 148.).

(10) Response to Argument

19. Regarding Appellant's arguments, Examiner believes that a vast majority of the Appellant's arguments are regarding specific terms used in the art and the claims, without consideration as to what the reference actually discloses. Much of what Appellant has argued alleges that Hull does not teach a particular feature, because a certain word is not used to disclose the feature, however, Examiner asserts that Hull teaches each and every limitation that Examiner has stated that Hull teaches, and that the lack of using identical language does not change the fact that Hull still reads on the claims. At the end of this section, Examiner has attached a table relating the terms used in the claims to the equivalent terms used in Hull, so that both the Board and the Appellant can see the relation between the claims and Hull.

20. Appellant's first argument, starting on Page 9 and continuing to Page 10, alleges that Hull does not disclose a decode unit, but instead states "instructions in bundles with lower memory addresses are considered to precede instructions in bundles with higher memory addresses". Examiner made this citation of Hull to disclose the limitation of sequentially fetching packets from memory, there is no explicit recitation of a decoder, however, as one of ordinary skill in the art would recognize, decoders are inherent in all computers which execute instructions (which is every computer). A decoder is a

necessary and inherent piece of hardware which takes a seemingly random collection of ones and zeros, and turns it into something that the computer can recognize as an instruction to operate upon. The fact that Hull operates on instructions necessitates a decoder, therefore, the issue is not whether or not a decoder exists, but whether or not the functionality claimed exists. Hull discloses that instructions are considered to be in a sequential order, determined by their memory address, therefore, Hull reads upon this particular claim limitation, the fact that Hull does not explicitly indicate a decoder does not in any way change the fact that a decoder is inherently present.

21. Appellant has further argued on Page 10 that "The cited portions relied on of Hull merely disclose different execution units, not processing channels as included in the processor architecture of Claim 1". This is a case of the Appellant focusing on the term used to describe an object, and not the object itself. The Appellant's own specification, on Page 4, clearly discloses that a processing channel is a collection of functional (execution) units, therefore, there is no apparent difference between a processing channel and a functional unit, the processing channel is simply a different term used to describe an execution unit to perform a function. The data processing channel is the collection of execution units which perform data processing, and the control processing channel is the collection of execution units responsible for jumps and branches, which is exactly how the Appellant in their own specification has defined the term, which is how Examiner has interpreted the claims, and is exactly what Hull discloses.

22. Appellant has further argued on Page 10 that the template field of Hull does not disclose the type of instruction packet, and argues that “An instruction slot is not an instruction packet as recited in Claim 1 but instead is part of an instruction bundle that includes multiple instruction slots”. Again, Appellant is arguing words, instead of what the words mean. Hull uses a template field to indicate what each type of instruction in an instruction bundle (packet) is, so that it can be directly to the appropriate functional units (processing channel). If the template field identifies what every instruction in the packet is, then it clearly has identified the type of packet. If the template field indicates that the packet only contains control instructions, then the packet is identified as such, and if the packet contains only data instructions, then it is also identified as such. Examiner can see absolutely no way that Hull does not read on this particular limitation, and asserts that again, Appellant is arguing based on the terms used to describe the invention being different from that of the art, while ignoring what the functionality of the term in the art actually is.

23. Appellant has further argued on Page 11 that Hull does not teach where instruction packets that have a plurality of only control instructions are caused to be executed sequentially on the control processing channel. However, Hull clearly teaches a packet which is comprised on only control instructions (Template "B", see Figure 4, it contains 3 branch instructions), and given that there are not three branch execution units, it would be impossible to execute them in parallel. And as Examiner has noted, and as Appellant has cited, Hull indicates that the instructions in a packet are executed

sequentially from 0, to 1, to 2, if they cannot be executed in parallel. Therefore, Hull appears to clearly read upon the claim limitations.

24. Appellant has also argued on Page 11 that Hull does not teach a decode unit (addressed above, as this is inherent), that causes instructions including at least one data processing instruction to be executed simultaneously on a data processing channel. However, this is exactly what Hull has stated in the same section that Appellant has cited, which states "As will be seen, the present invention provides a processor capable of simultaneously executing a plurality of sequential instructions", which Examiner notes is the entire purpose of the invention. Therefore, Appellant appears to have cited the part of Hull which conclusively shows how Hull reads upon the claim limitations, therefore, Examiner does not see how it is reasonable to argue that Hull does not teach the claim limitations, when Hull clearly and definitively has recited the limitations in the claim, as pointed out by the Appellant.

25. Finally, Appellant argues on Pages 12-13 that Hennessy and Hull would not be obvious to combine, because allegedly, if Hull were to use a variable length instruction set, none of the waste or inefficiency which Hull corrects for in his invention would exist. However, Examiner believes that this is not a reasonable position to take. The point of the Hull invention, as described in Column 1, Lines 64-67, is to remove a large number of multiplexors and wiring to route instructions to their various execution units. Examiner notes that Hull refers to this being an issue with a RISC machine, however, the same

issue exists with variable length instructions. Just because an instruction has a variable length, instead of a fixed length, does not mean that the system somehow can route instructions to the appropriate execution units with fewer wires or multiplexors, the length of the instruction has nothing to do with the ability to send it to the appropriate functional unit. Hull's invention allows for the machine to more easily route instructions to the proper location by minimizing wiring and multiplexors, which is something that would be advantageous for any system, and Hull only mentions a RISC machine by name because that is the system he is using, but in no way indicates that the problem somehow does not exist for a variable length instruction set.

26. Applicant has further argued on Page 13 that the fact that the Examiner has used a secondary reference to make a 103 rejection of the claims somehow is proof that the claims are non-obvious, however, Examiner does not see how this is a relevant argument, as the Appellant appears to be ignoring the entire 103 statute in regards to obviousness, if Examiner was not using a secondary reference, then the claim would have been rejected under 102, and the issue at hand would be novelty, not obviousness. The fact that the Examiner used a textbook as a secondary reference, if anything, should demonstrate exactly how well known and obvious the subject matter is, not the other way around.

27. Examiner notes that the remainder of the arguments presented in the brief appear to be identical to the arguments previously addressed, and therefore Examiner will not address them for brevities sake, and instead refers to his above comments.

28. The following is a table to compare the terms used in the claims and the terms used in the Hull reference where the Appellant appears to have not recognized or understood the mapping, and Examiner provides this table to make it clear what Examiner is referring to when mapping the claims to the reference.

Instant Application	Hull Reference
Instruction Packet	Instruction Bundle
Processing Channel	Execution unit (or groups thereof)
Control Instruction	Branch Instruction
Identification Bits	Template Field

29. Examiner notes the IDS filed 1/21/2010, and has entered it into the record, however, has not considered it, because it is not in compliance with 1.97, which states that for any IDS filed after a final action, both a statement as specified in section (e) must accompany the IDS, as well as the fee set forth in 1.17(p). No fee has been paid, therefore, the IDS has not been considered.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Robert Fennema

/Robert Fennema/

Conferees:

/Kevin L Ellis/
Supervisory Patent Examiner, Art Unit 2187

Eddie Chan

/Eddie P Chan/

Supervisory Patent Examiner, Art Unit 2183